AD NUMBER

AD838959

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; JUN 1968. Other requests shall be referred to Naval Postgraduate School, Code 023, Monterey, CA 93940. This document contains export-controlled technical data.

AUTHORITY

USNPS ltr, 16 Nov 1971

UNITED STATES
NAVAL POSTGRADUATE SCHOOL

THESIS

COMPUTATION OF MAXIMUM

FLOWS IN NETWORKS

by

William Charles Burns

June 1968

Code 123, Monterey, Calif 93940

55

COMPUTATION OF MAXIMUM FLOWS IN NETWORKS

by

William Charles Burns
Captain, United States Army
B.S., Military Academy, 1962

Submitted in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN OPERATIONS RESEARCH
from the
NAVAL POSTGRADUATE SCHOOL
June 1968

Signature of Author William C. Burns

Approved by Arnold Kaufmann
Thesis Advisor

Chairman, Department of Operations Analysis

W. F. Koehler for R. F. Rinehart
Academic Dean

ABSTRACT

A review of the current theory and methods for the computation of maximum flow in networks is presented along with a simplified method for determination of a feasible flow in networks with upper and lower bounded arcs. A computational procedure is presented which is used to calculate the maximum flow for a general network. The network is reduced to an equivalent basic network. An associated network is used to compute a feasible, then the maximum flow for the basic network. A computer program is included for use in computation of maximal flows in large networks.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

The growing uses for mathematical programming models in transportation and communication networks have prompted study in the development of computational methods to determine maximal network flows. The subject of this thesis is the mathematical problem of determining maximal steady state flows in networks which are subject to capacity limitations on the arcs and nodes of the network.

An efficient and widely used method to compute maximal flow was developed by Ford and Fulkerson [7]. This method along with a review of the current theory will be presented in Chapter II in order to give the reader the theoretical foundation upon which the remainder of the thesis is based. Chapter III introduces a new method which utilizes the Ford and Fulkerson algorithm to compute the maximal flow in networks with non-zero lower bound limitations on the flow in the arcs. Since many networks of interest are more general in nature, the generalized network is introduced in Chapter IV along with a computational procedure for the determination of maximal flow. To illustrate the techniques proposed, an example is included. The procedure can be used to solve small problems by hand, but the amount of work increases rapidly with the size of the network. Therefore, a computer program which may be used to solve larger problems is included in the appendices. A brief description of the program is found in Chapter IV.

Definitions and Symbols

In order to establish a common understanding as to the exact meaning of various terms and symbols used to present the material of this thesis, an initial set of definitions is presented. Other terms will be defined throughout the material as needed to facilitate the presentation.

Define a network, G, as a graph which can be represented on a plane in such a way that the set, N, of nodes are distinct points and the set, A, of arcs are simple curves which connect two distinct nodes. Furthermore, no two arcs can meet except at the nodes which are their extremities. The graph contains no loops.

Define $\omega^{-}(N)$ to be the set of arcs or flows incident to a set of nodes, N, and $\omega^{+}(N)$ to be the set of arcs or flows incident from N.

A directed arc, $a_{i,j}$, is defined as an arc incident from a node i and incident to a node j.

An undirected arc connects node i and node j without having an orientation.

Flow can be thought of as a value of the steady state rate of movement of a homogeneous commodity along a path or channel. Positive flow in a directed arc, $a_{i,j}$, will move with the orientation of the arc and will be denoted by the symbol, $y_{i,j}$. Positive flow in an undirected arc, denoted $\bar{y}_{i,j}$, may move in either direction but not in opposite directions at the same time.

To each arc, $a_{i,j} \subset A$, there will be associated two real numbers, $b_{i,j}$, and $c_{i,j}$, which represent, respectfully, the minimum and maximum allowable flow in that arc. The bounds associated with undirected arcs, $\bar{a}_{i,j}$, are denoted $\bar{b}_{i,j}$, and $\bar{c}_{i,j}$.

An arc is <u>saturated</u> if $y_{i,j} = c_{i,j}$.

Define <u>Y</u> as the set of flows in a network G.

Initially, the capacities of the nodes will be assumed to be infinite. The nodes will be classified according to the value of the difference of the sum of the flows into the node minus the value of the sum of the flows out of the node. A node is designated as a <u>source node</u> if this difference is negative, and as a <u>sink node</u> if this difference is positive. For all intermediary nodes, the sum of the flows into a node must equal the sum of the flows out. The set of source, intermediary and sink nodes will be denoted as S, I, and T respectfully.

Define <u>F</u> as the value of the flow in a network.

$$F = \sum_{y_{1,t} \underline{c} \omega^-(T)} y_{i,t}$$

A flow is feasible if and only if:

(1) $b_{i,j} \leq y_{i,j} \leq c_{i,j} \qquad \forall\, a_{i,j} \underline{c} A$

(2) $$F - \sum_{y_{s,j} \underline{c} \omega^+(S)} y_{s,j} = 0$$

$$\sum_{y_{i,t} \underline{c} \omega^-(T)} y_{i,t} - F = 0$$

(3) $$\sum_{y_{i,j} \underline{c} \omega^-(i)} y_{j,i} - \sum_{y_{i,j} \underline{c} \omega^+(j)} y_{i,j} = 0 \qquad i,j \underline{c} I$$

Define a <u>chain</u> as a sequence of arcs such that each arc, $a_{i,j}$ $i,j \underline{c} I$, is connected to an arc $a_{k,i}$ at node i and to arc $a_{j,l}$ at node j.

A <u>cycle</u> is a sequence of arcs where:

(1) The sequence is a chain.
(2) The sequence does not use the same arc twice.
(3) The initial and terminal nodes of the chain coincide.

A cut is defined as any set of directed arcs containing at least one arc from every chain of positive capacity joining the source to the sink. The value of the cut is equal to the sum of the capacities of the arcs of the cuts.

Basic Network

Much of the initial development of the material in this thesis deals with computation of flows in simple or basic networks. Therefore, for the purpose of this paper, a basic network, G, is defined as a network which has the following properties.

1. The network has a return arc, $a_{n,1}$, which is added to the network for computational purposes only.
2. Node 1 has only one arc, $a_{n,1}$, incident to it. Node 1 is designated as the source. In the basic network there is only one source node.
3. Node n has only one arc, $a_{n,1}$, incident from it. Node n is designated as the sink. In the basic network there is only one sink node.
4. For each arc, $a_{i,j}$, the associated $c_{i,j}$, is positive and $b_{i,j} = 0$.
5. Two distinct nodes (i,j) may be connected by only one directed arc. The network contains no undirected arcs.

CHAPTER II

REVIEW OF THE LITERATURE

In order to acquaint the reader with the subject of maximal flow in networks, a review of the current theory and primary combinational techniques is presented in this chapter.

Basic Network Theory

A great deal of theoretical work in networks has already been done. A few of the relevant theorems are presented here in order to provide justification and a rationale for the combinatorial techniques and algorithms that are presented in later sections.

Theorem 1. Opposite directed flows on the same arc cancel [4]. This theorem means that given an undirected arc, $\bar{a}_{i,j}$, with flows $\bar{y}_{i,j} \geq 0$ and $\bar{y}_{j,i} \geq 0$, these two flows can be replaced by the flows $y'_{i,j}$ and $y'_{j,i}$

$$y'_{i,j} = \bar{y}_{i,j} - \min(\bar{y}_{i,j}, \bar{y}_{j,i})$$

Theorem 2. A set Y with F>0, satisfying the capacity constraints and node conservation equations can be decomposed into a set of positive chain flows from the source to the sink and a set of circular flows such that the direction of positive flows in any common arc is the same for all chains [4].

Theorem 3. There exists a positive flow from the source to the sink if there exists at least one chain of arcs with positive capacity from the source to the sink [4].

Theorem 4. A flow F° is maximal if and only if the maximal flow is zero in a second network formed by replacing $c_{i,j}$ by $c_{i,j} - y_{i,j}$ $\forall$ $a_{i,j} \subseteq$ A.

Theorem 5. Given any partition of nodes into two sets, where the first set includes the source node and the second set the sink, then a feasible solution F is maximal, if every arc, $a_{i,j}$, that joins a node in the first set to a node of the second is saturated [4].

Theorem 6. The maximum flow in a network is equal to the minimum cut.

Tree Method

Dantzig [4] has developed a systematic procedure for computing the maximal flow in a network by using chains connecting the source and the sink of the network. A knowledge of the tree method will give a better understanding of the Ford and Fulkerson algorithm which is presented in the next section because the two methods are identical in principle.

A tree [1] is defined as a connected graph with at least two nodes and no cycles.

Consider a basic network such that:

$$b_{i,j} = 0 \ , \ c_{i,j} \geq 0 \quad \forall \ a_{i,j} \subseteq A$$

The arcs may be directed or undirected. Initially all arcs are unmarked and the feasible flow is zero. In order to determine the maximum flow:

1. Create a tree such that the arcs are unsaturated.
2. Select two sub trees - one T_1, branching out from the source, node 1, and the other, T_n, branching out from the sink, n, such that each node is reached by just one arc of the tree.
3. Choose one unsaturated arc which connects the two trees. Thus, there will be just one chain from node 1 to node n. If none can be found, go to step 7.

4. Determine the amount that the flow can be increased along the chain from node 1 to node n, by the following procedure. Determine the amount that the flow can be increased, ΔX, where $\Delta X = \min (c_{i,j} - y_{i,j}, y_{k,l})$. Flow along the chain directed toward the sink is designated $y_{i,j}$ and flow directed toward the source along the chain is designated $y_{k,l}$. There are two cases:

   (a) $\Delta X > 0$, increase the flow by ΔX along the chain, go to next step.

   (b) $\Delta X = 0$. This means one arc in the chain is saturated by a flow directed toward the sink, go to next step.

5. Mark one of the saturated arcs.

6. Eliminate the saturated arc from the chain. Thus, there are two sub-trees. Return to step 3.

7. Check all marked arcs joining T_1 to T_n. If all these arcs have flow from T_1 to T_n then the optimum has been reached. If any arc has flow from T_n to T_1, use this arc to connect T_1 to T_n and proceed with step 4.

Ford and Fulkerson Algorithm

Given the basic network, the most direct method developed to date of computing the maximum flow is the algorithm developed by Ford and Fulkerson. This method is the one most cited in the literature concerning maximal network flow. The procedure is outlined below.

1. Begin with a feasible flow in 0 in all arcs.
2. Label each node as follows:
   a. Label node 1, with the number 1.

b. If node i is labeled and node j is not labeled, label node j with the number (i) if:

(1) $a_{i,j}$ exists and $y_{i,j} < c_{i,j}$

(2) $a_{j,i}$ exists and $y_{j,i} > 0$

c. If the sink, node n, becomes labeled by this procedure, then the flow from node 1 to node n can be increased.

3. If node n is labeled, construct a simple chain from node 1 to node n by back tracking from n to 1 using the labels on the nodes.

a. The chain created will have the property that all arcs, $a_{i,j}$ on the chain directed toward the sink have a flow $0 \leq y_{i,j} \leq c_{i,j}$ and all arcs $a_{1,k}$ directed toward the source will have a flow $y_{1,k} > 0$.

b. The flow can be increased by an amount ΔX where:

$\Delta X = \min(c_{i,j} - y_{i,j}, y_{1,k})$ $\forall$ $a_{i,j}$, $a_{1,k}$ in chain.

4. Increase the flow along the chain by an amount ΔX and return to step 2.

5. This procedure is repeated until the sink cannot be labeled. Flow is maximized when this is the case.

Now consider the more general problem in which the lower bounds on the directed arcs are no longer assumed to be zero. Thus the flow in each arc is bounded both above and below.

Such that: $-\infty \leq b_{i,j} \leq y_{i,j} \leq c_{i,j} \leq +\infty$ $\quad a_{i,j} \subseteq A$

Theorem 7: In the case of networks with lower bounds on the arcs, C is defined as the subset of N such that if node 1 $\notin$ C the node n $\notin$ C or if node 1 $\subseteq$ C then node n $\subseteq$ C.

A flow $y_{i,j}$ exists and is feasible if and only if:

1. $b_{i,j} \leq y_{i,j} \leq c_{i,j} \quad \forall\, a_{i,j} \subseteq A$

2. $$\sum_{a_{i,j} \subseteq \omega^-(C)} c_{i,j} \geq \sum_{a_{i,j} \subseteq \omega^+(C)} b_{i,j}$$

This more general type of problem can be satisfactorily worked by using the Ford and Fulkerson algorithm if a feasible flow can be found.

The Ford and Fulkerson algorithm can be used to solve for a maximum flow in an associated network G' which is derived from the original network, G. This method is outlined below.

1. Create an associated network G' such that G' contains the nodes and arcs of the original network G with arc capacities $c'_{i,j} = c_{i,j} - b_{i,j}$. The lower bound, $b'_{i,j}$ equals zero.
2. Add a new source, labeled 0, and a new sink, labeled m, $(m = n + 1)$, to G'.
   a. If the arc, $a_{i,j}$, exists, then construct two new arcs according to the value of the lower bound, $b_{i,j}$. If $b_{i,j} > 0$, then construct the arcs, $a'_{i,m}$ of capacity $c'_{i,m} = b_{i,j}$ and $a'_{o,j}$ with capacity $c'_{o,j} = b_{i,j}$. If $b_{i,j} < 0$, then construct the arcs, $a'_{o,i}$ with capacity $c'_{o,i} = -b_{i,j}$ and $a'_{j,m}$ with capacity $c'_{j,m} = -b_{i,j}$.
   b. The Ford and Fulkerson algorithm can then be used to determine the maximum flow in G' from node 0 to node m.
3. Two cases can result from step 2b.
   a. Case one: If the flow in each arc, $a'_{o,j}$ and $a'_{j,m}$

$j \subseteq N'$, equals the capacity of each arc, then the flow is feasible for the network G and the problem has a solution.

b. Case two: If for any arc, $a'_{o,j}$ or $a'_{j,m}$ $j \subseteq A'$, the flow in that arc does not equal the capacity of the arc, the problem has no solution.

4. If a feasible flow for G can be found by the above procedure, transfer the flow from G' to G by using the following transformation:

$$y_{i,j} = y'_{i,j} + b_{i,j} \quad \forall\, a_{i,j} \subseteq A$$

5. Ford and Fulkerson's algorithm is then used to determine the maximum flow using the feasible flow as a starting basis.

CHAPTER III

A SIMPLIFIED METHOD FOR COMPUTATION OF MAXIMAL FLOW

To compute a maximum flow in networks with non-zero lower bounded arcs, an initial feasible flow must be determined. The method presented in the last chapter produces a complex associated network, G', even with a relatively simple network, G. The principle advantage of the method presented here [10] is that an initial feasible flow can be determined by use of a more simple network, G', than in the previous method.

Proposed Algorithm

The development of the algorithm is based upon a consequence of the flow balance equations. In order to provide a rationale for the procedures presented, the development of the associated network will be presented now.

Given a basic network G where the flow in each arc is bounded as follows:

$$-\infty \leq b_{i,j} \leq y_{i,j} \leq c_{i,j} + \infty \qquad a_{i,j} \in A$$

The problem of determining the maximum flow in G can be stated as follows:

Maximize: $y_{n,1}$

Subject to: $$\sum_{y_{j,i} \in \omega^-(i)} y_{j,i} - \sum_{y_{i,j} \in \omega^+(i)} y_{i,j} = 0 \quad \forall\, i \in N$$

$$b_{i,j} \leq y_{i,j} \leq c_{i,j} \qquad \forall\, a_{i,j} \in A$$

Now transform the entire network G into an associated network G' using the change of variables:

$$y_{i,j} = y'_{i,j} + b_{i,j} \qquad \forall\, a_{i,j} \in A$$

the problem then becomes:

Maximize: $y'_{n,1}$

Subject to:

$$y'_{i,j} \sum_{\underline{c}\ \omega^-(i)} y'_{j,i} - y'_{i,j} \sum_{\underline{c}\ \omega^+(i)} y'_{i,j} + b_{j,i} \sum_{\underline{c}\ \omega^-(i)} b_{j,i} - b_{i,j} \sum_{\underline{c}\ \omega^+(i)} b_{i,j} = 0 \quad \forall\ i \underline{c} N$$

and: $0 \leq y'_{i,j} \leq c'_{i,j} = c_{i,j} - b_{i,j}$

Let $\sum_{b_{j,i}\ \underline{c}\ \omega^-(i)} b_{j,i} - \sum_{b_{i,j}\ \underline{c}\ \omega^+(i)} b_{i,j} = d_i \qquad i \underline{c} N$

Note two things. First the sum, $\sum_{i \underline{c} N} d_i$, is equal to zero. Second, if a flow of $-d_i$ was passed through each node, the problem would be exactly the problem of flow maximization in a basic network. Hence, a technique similar to Berge's [2] can be used in which new nodes 0 and m = n+1 are added to the network G'. Therefore, only a maximum of one arc per node is added to G' in this method.

Procedure

Using the results of the previous section the solution to the maximal flow in the above network can be determined using the following procedure.

1. Add a new source and sink node, denoted 0 and m (m = n + 1), respectfully to G'.

2. The arcs $a'_{0,i}$ and $a'_{j,m}$ are added as follows. If $d_i > 0$, then an arc, $a'_{0,i}$, is constructed with capacity d_i. If $d_i < 0$, an arc, $a'_{j,m}$, is constructed with capacity $-d_i$. If $d_i = 0$, no arc is added to G' for node i.

3. The Ford and Fulkerson algorithm is then used to maximize flow in G' from node 0 to node m. Again as in the previous method this maximum flow is equivalent to a feasible flow in the network G if and only if:

$$y'_{o,i} = d_i \qquad \forall\, i \ni d > 0$$

$$y'_{j,m} = d_i \qquad \forall\, i \ni d < 0$$

4. If the problem is solvable continue by eliminating all arcs incident from node 0 and incident to node m. Node 1 and node n are now the source and sink respectfully for G'.

5. Using the flow generated in step 3 as a starting basis, maximize flow from node n to node 1 by again using the Ford and Fulkerson Algorithm.

6. Upon completion of step 5, transfer flow from G' to G by use of the transformation equation:

$$y_{i,j} = y'_{i,j} + b_{i,j} \qquad \forall\, a_{i,j} \subseteq A$$

The maximum flow in G is equal to $y_{n,1}$.

CHAPTER IV

PROCEDURE FOR THE SOLUTION TO THE GENERALIZED MAXIMUM FLOW PROBLEM

A proposed methology for the computation of a solution to the generalized maximal flow type problem as defined below is presented. The generalized network is first reduced to a basic network and then the technique presented in Chapter III is used to solve for the equivalent maximum flow in the basic network. An example is presented in order to demonstrate the techniques discussed in this chapter. A description of the computer program contained in Appendix A for use in solving maximal flow problems for large networks is included.

The Generalized Network

Generally maximal flow solutions are required for networks which do not fit the restrictive definition of a basic network. For the purpose of this thesis the generalized network is defined as a modified basic network with a set of arcs A, and a set of nodes N, such that:

1. Each set $\omega^-(i)$ and $\omega^+(i)$ can contain any number of directed and undirected arcs.
2. Let D be a subset of A such that if and only if the arc $a_{i,j}$ is directed it is a member of D.
3. Let U be a subset of A such that if and only if the arc $\bar{a}_{i,j}$ is undirected it is a member of U.
4. K is a subset of N such that if the node i is bounded it is a member of K. To the bounded node, i, we associate two real numbers, b_i, the lower bound, and, c_i, the upper

bound such that the total flow through the node i, denoted y_i, is bounded:

$$b_i \leq y_i \leq c_i \qquad i \subseteq K$$

5. All nodes, j, not bounded belong to the subset l, where $L \subseteq N$.
6. The network may or may not contain multiple sources or sinks, but shall contain at least one source and one sink.

Procedure for Reduction of the Generalized Flow Problem

In order to solve for the maximal flow in the generalized network by the techniques presented in Chapter III, the network must be transformed into an equivalent basic network.

The proposed methods for reduction are based upon results of the general theorems presented in Chapter II. References cited give further proof of some of the methods used.

First, consider the case in which the generalized network, G, contains both directed and undirected arcs. This network can be simplified by replacing each undirected arc by two directed arcs as follows:

Given an undirected arc $\bar{a}_{i,j} \subseteq U$

Replace $\bar{a}_{i,j}$ by $a_{i,j}$ and $a_{j,i} \subseteq D$

Where

$$0 \leq y_{i,j} \leq c_{i,j} = \bar{c}_{i,j}$$

$$0 \leq y_{j,i} \leq c_{j,i} = \bar{c}_{i,j}$$

Upon reaching a solution for the maximal flow, the orientation by the flow in the undirected arcs can immediately be determined from the values of $y_{j,i}$ and $y_{i,j}$ since one flow will be zero.

Now consider a network in which two distinct nodes, i, j, are connected by more than one arc. Successive arcs, flows, and bounds are denoted by prime symbols. For example if there are m arcs from node i to node j, the k^{th} arc would be denoted $a_{i,j}^{(k)}$. It will be assumed that each arc has both upper and lower bounds.

$$b_{i,j}^{(k)} \leq y_{i,j}^{(k)} \leq c_{i,j}^{(k)} \qquad k = 1,2,\ldots m$$

The set of arcs $a_{i,j}^{(k)}$ $k = (1,\ldots m)$ may be replaced by a single arc $a_{i,j}$, where; $c_{i,j} = \sum_{k=1}^{m} c_{i,j}^{(k)}$ $\qquad b_{i,j} = \sum_{k=1}^{m} b_{i,j}^{(k)}$

Next assume a pair of nodes (i,j) are connected by two oppositely directed arcs $a_{i,j}$ and $a_{j,i} \subseteq D$.

The pair of arcs may be replaced by a single directed arc in either direction as follows: If $c_{i,j} \geq b_{j,i}$

Replace $a_{i,j}$ and $a_{j,i}$ by $a_{i,j}$ such that:

$$c'_{i,j} = c_{i,j} - b_{i,j}$$

$$b'_{i,j} = b_{i,j} - c_{i,j}$$

Otherwise orient the arc from j to i.

If a network has multiple sources and, or sinks, merely create a single source and, or single sink. Then construct one directed arc of infinite capacity from the new source to each old source and one directed arc of infinite capacity from each sink to the new sink [8].

If the flow through the nodes is restricted to lie between two values, the character of the maximum flow problem itself remains unaltered. Assume we are given a network in which the flow is subject to both arc and node capacities, where all arcs are directed or have been converted to directed arcs by the previous techniques. This network can be reduced to the basic network maximum flow problem by the following procedure. Expand the network such that for each bounded

node (i) $\subseteq$ K, two new unbounded nodes (i', i'') are created, so that the single directed arc, $a_{i'',j''}$, joins the two new nodes with the upper and lower bounds associated with the bounded node in the reduced network. To each arc, $a_{k,i}$, create an arc $a'_{k,i'}$ in G' and to each arc $a_{i,j}$ in G create an arc $a_{i'',j}$ in G'. The bounds on the other arcs remain the same.

Example Using the Simplified Technique

Given the generalized network shown in Figure 1, we are required to compute the maximal flow from node 2 and 3 to node 10. Or;

$$\text{Maximize } F = \sum_{y_{i,n} \subseteq \omega^-(n)} y_{i,n} = y_{5,10} + y_{7,10} + y_{6,10}$$

The flow in each arc is bounded as follows:

$3 \leq y_{2,4} \leq 8$ $\quad 0 \leq y_{2,6} \leq 4$ $\quad 3 \leq y_{3,4} \leq 7$

$5 \leq \underset{(1)}{y_{3,5}} \leq 10$ $\quad 3 \leq \underset{(2)}{y_{4,5}} \leq 8$ $\quad 0 \leq \bar{y}_{4,5} \leq 5$

$0 \leq y_{4,6} \leq 3$ $\quad 1 \leq y_{4,6} \leq 2$ $\quad 0 \leq \bar{y}_{4,6} \leq 2$

$3 < y_{5,4} \leq 8$ $\quad 0 \leq y_{5,6} \leq 8$ $\quad 3 \leq y_{5,7} \leq 7$

$2 \leq y_{5,10} \leq 5$ $\quad 1 \leq \underset{(1)}{y_{6,7}} \leq 3$ $\quad 3 \leq \underset{(2)}{y_{6,10}} \leq 9$

$1 \leq y_{7,5} \leq 4$ $\quad 1 \leq y_{7,6} \leq 3$ $\quad 2 \leq y_{7,6} \leq 5$

$4 \leq y_{7,10} \leq 10$

The flow in all nodes except 5 and 6 is unrestricted. The flow in node 5,6 is restricted as follows,

$4 \leq y_5 \leq 15$ $\quad 4 \leq y_6 \leq 10$

This flow problem can be simplified by using the techniques presented in the last section. The procedure is outlined by ordered steps. Although many steps could be combined for this simple

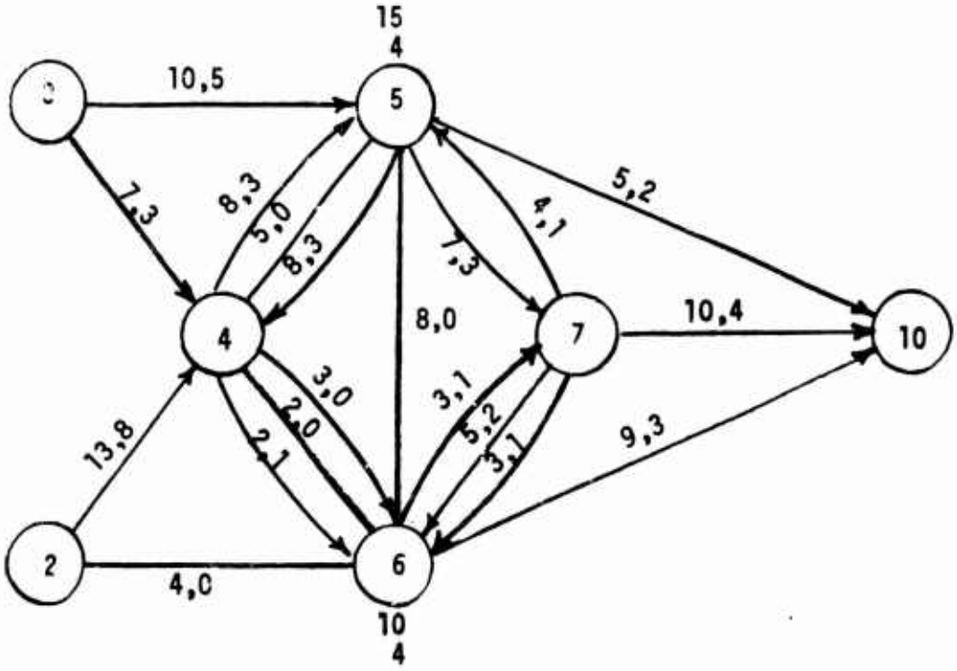


FIGURE 1

GENERALIZED NETWORK G WITH UPPER
AND LOWER BOUNDS ON ARCS AND NODES

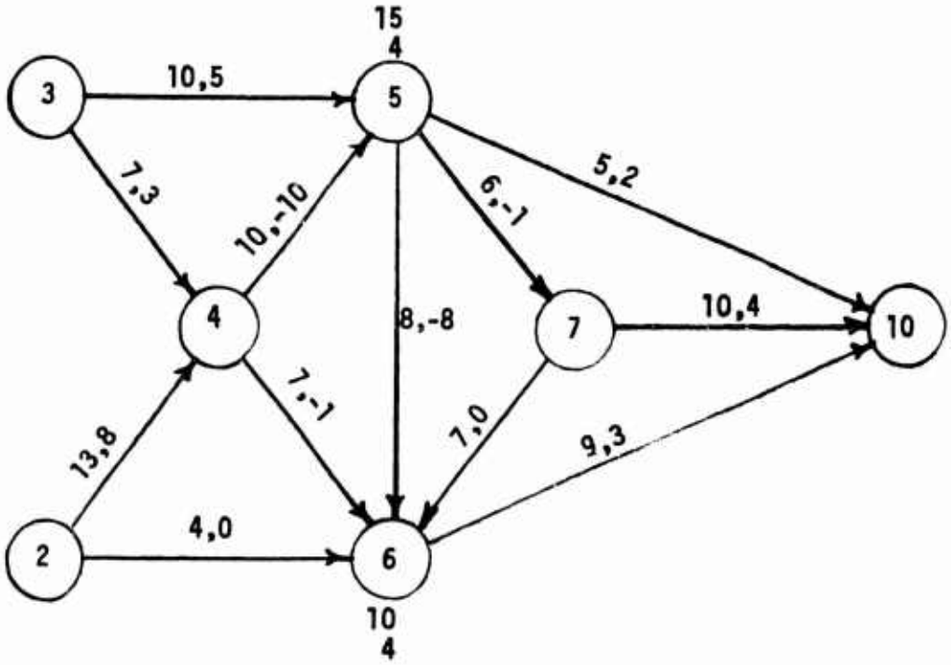


FIGURE 2

GENERALIZED NETWORK WITH SINGLE
ARCS REPLACING MULTIPLE ARCS

example, it is felt that for hand and computer computation this is the most efficient procedure.

1. Add the single source node, denoted as 1, to the network. In addition, add the arcs $a_{1,2}$ and $a_{1,3}$ to the network such that:

   $0 \leq y_{1,2} \leq +\infty$

   $0 \leq y_{1,3} \leq +\infty$

2. Replace each undirected arc $\bar{a}_{i,j}$ with upper bound $\bar{c}_{i,j} \geq 0$ by two directed arcs $a_{i,j}$ and $a_{j,i}$.

   Therefore:

   $\bar{a}_{4,5}$ becomes $a^{(2)}_{4,5}$ and $a^{(2)}_{5,4}$ $\quad 0 \leq y^{(2)}_{4,5} \leq 5,\ 0 \leq y^{(2)}_{5,4} \leq 5$

   $\bar{a}_{4,6}$ becomes $a^{(3)}_{4,6}$ and $a_{6,4}$ $\quad 0 \leq y^{(3)}_{4,6} \leq 2,\ 0 \leq y_{6,4} \leq 2$

   $\bar{a}_{5,6}$ becomes $a_{5,6}$ and $a_{6,5}$ $\quad 0 \leq y_{5,6} \leq 8,\ 0 \leq y_{6,5} \leq 8$

3. Combine all arcs joining each pair of arcs (i,j) according to the following procedure.

   a. Combine all arcs which are similarly directed, into a single arc. Thus the arcs $a^{(1)}_{4,6}$, $a^{(2)}_{4,6}$, $a^{(3)}_{4,6}$ become $a_{4,6}$ ∋

   $c_{4,6} = c^{(1)}_{4,6} + c^{(2)}_{4,6} + c^{(3)}_{4,6} = 3 + 2 + 2 = 7$

   $b_{4,6} = b^{(1)}_{4,6} + b^{(2)}_{4,6} + b^{(3)}_{4,6} = 1 + 0 + 0 = 1$

   Similarly form the arcs $a_{4,5}$, $a_{5,4}$ and $a_{7,6}$ ∋

   $c_{4,5} = 13 \qquad b_{4,5} = 3$

   $c_{5,4} = 13 \qquad b_{5,4} = 3$

   $c_{7,6} = 8 \qquad b_{7,6} = 8$

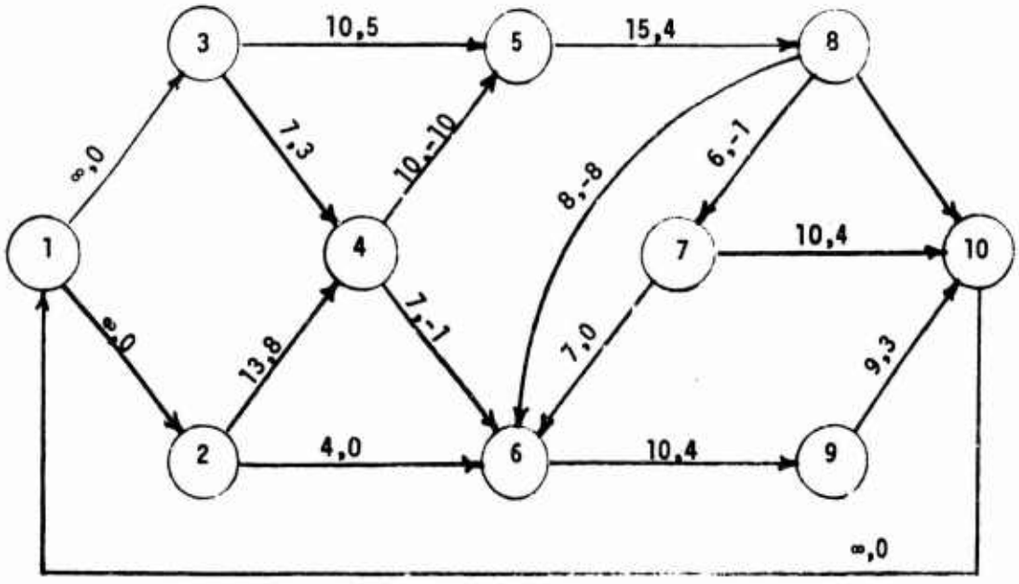


FIGURE 3
EQUIVALENT BASIC NETWORK

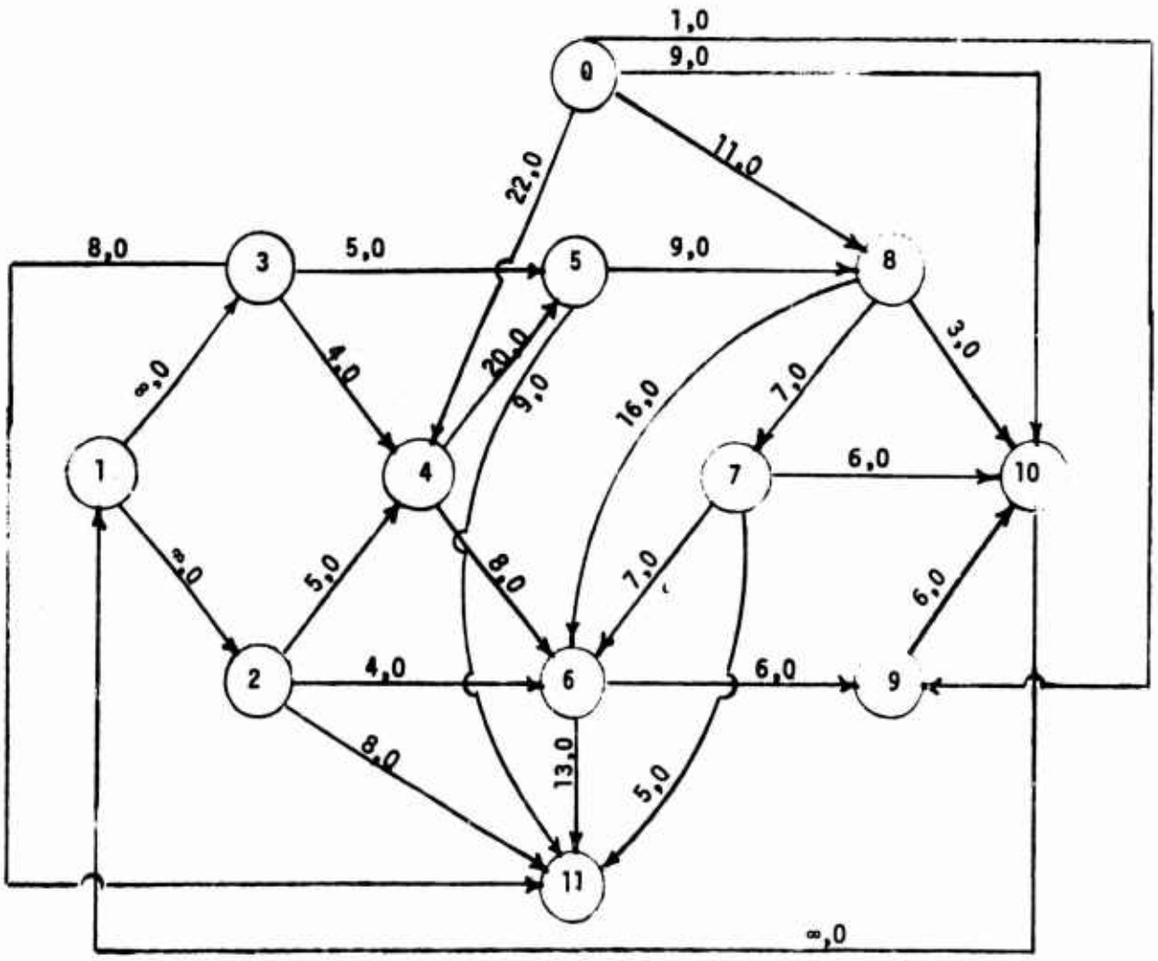


FIGURE 4
TRANSFORMED NETWORK G'

b. Transform each pair of arcs which are directed in opposite directions between each pair of nodes into a single arc.

Thus: $a_{4,5}$ and $a_{5,4}$ become $a_{4,5}$ ∋

$c_{4,5} = c_{4,5} - b_{5,4} = 13 - 3 = 10$

$b_{4,5} = b_{4,5} - c_{5,4} = 3 - 13 = -10$

Similarly the pair of arcs $a_{4,6}$ and $a_{6,4}$; $a_{5,6}$ and $a_{6,5}$; $a_{5,7}$ and $a_{7,5}$; $a_{7,6}$ and $a_{6,7}$ are combined. The simplified network is shown in Figure 2.

3. Complete the transformation to the basic network by the following procedure.

   a. Expand the network so that all nodes have infinite capacity for flow as outlined earlier. Thus for nodes 5, and 6 the new arcs $a_{5,8}$ and $a_{6,9}$ are created and the equivalent network shown in Figure 3 is established.

   b. Add the return arc $a_{10,1}$ for computational purposes.

4. Transform the network G to G' by using the change of variables equation $y_{i,j} = y'_{i,j} + b_{i,j}$ $a_{i,j} \subseteq A$

$y_{10,1} = y'_{10,1}$ $y_{1,2} = y'_{1,2}$ $y_{1,3} = y'_{1,3}$

$y_{2,4} = y'_{2,4} + 8$ $y_{2,6} = y'_{2,6}$ $y_{3,4} = y'_{3,4} + 3$

$y_{3,5} = y'_{3,5} + 5$ $y_{4,5} = y'_{4,5} - 10$ $y_{4,6} = y'_{4,6} - 1$

$y_{5,8} = y'_{5,8} + 4$ $y_{6,9} = y'_{6,9} + 4$ $y_{7,6} = y'_{7,6}$

$y_{7,10} = y'_{7,10} + 4$ $y_{8,6} = y'_{8,6} - 8$ $y_{8,7} = y'_{8,7} - 1$

$y_{8,10} = y'_{8,10} + 2$ $y_{9,10} = y'_{9,10} + 3$

The equivalent problem is obtained:

Maximize $y'_{10,1}$ with the node flow satisfying the balance equations as follows:

Node

1	$y'_{10,1} - y'_{1,3} - y'_{1,2}$	$= 0$
2	$y'_{1,2} - y'_{2,4} - y'_{2,6} - 8$	$= 0$
3	$y'_{1,3} - y'_{3,4} - y'_{3,5} - 8$	$= 0$
4	$y'_{3,4} + y'_{2,4} - y'_{4,5} - y'_{4,6} + 22$	$= 0$
5	$y'_{3,5} + y'_{4,5} - y'_{5,8} - 9$	$= 0$
6	$y'_{2,6} + y'_{4,6} + y'_{7,6} + y'_{8,6} - y'_{6,9} - 13$	$= 0$
7	$y'_{8,7} - y'_{7,6} - y'_{7,10} - 5$	$= 0$
8	$y'_{5,8} - y'_{8,6} - y'_{8,7} - y'_{8,10} + 11$	$= 0$
9	$y'_{6,9} - y'_{9,10} + 1$	$= 0$
10	$y'_{8,10} + y'_{7,10} + y'_{9,10} - y'_{10,1} + 9$	$= 0$

Therefore the values of the remainder d_i are immediately seen to be as follows:

$$d_1 = 0 \quad d_2 = -8 \quad d_3 = -8 \quad d_4 = +22$$

$$d_5 = -9 \quad d_6 = -13 \quad d_7 = -5 \quad d_8 = +11 \quad d_9 = +1 \quad d_{10} = +9$$

Using the values of d_i, $i = 1,\ldots,10$, as the capacities, the new source, sink and arcs are added to obtain the equivalent network G' shown in Figure 4.

5. Then using the Ford Fulkerson algorithm to obtain a feasible flow in G the following values are obtained for the flow in G'.

$y'_{1,2} = 12$ $y'_{1,3} = 8$ $y'_{2,4} = 0$ $y'_{2,6} = 4$

$y'_{3,4} = 0$ $y'_{3,5} = 0$ $y'_{4,5} = 14$ $y'_{4,6} = 8$

$y'_{5,8} = 5$ $y'_{6,9} = 5$ $y'_{7,6} = 0$ $y'_{7,10} = 2$

$y'_{8,6} = 6$ $y'_{8,7} = 7$ $y'_{8,10} = 3$ $y'_{9,10} = 6$

$y'_{10,1} = 20$

6. Note that the flow in all sink arcs and source arcs is at arc capacity, therefore, a feasible flow exists. Dropping nodes 0 and 11 plus the associated arcs, maximize flow from node 1 to node 10 using the Ford Fulkerson algorithm. In this case the feasible flow computed is the maximum flow. Then, using the change of variables equations the resulting maximal flow on G is determined. Finally G is simplified by eliminating arcs $a_{5,8}$ and $a_{6,9}$. The maximal flow is 20. The flow in each arc is as follows:

   $y_{2,4} = 8$ $y_{2,6} = 4$ $y_{3,4} = 3$ $y_{3,5} = 5$

   $y_{4,5} = 7$ $y_{4,6} = 7$ $y_{5,4} = 3$ $y_{5,7} = 7$

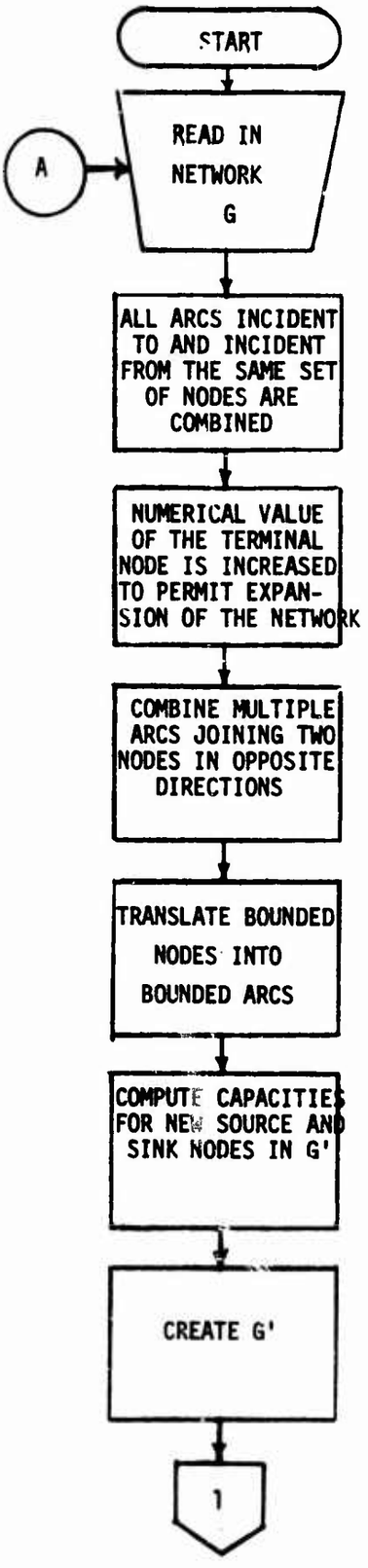
   $y_{5,10} = 5$ $y_{6,4} = 0$ $y_{6,5} = 2$ $y_{6,7} = 3$

   $y_{6,10} = 9$ $y_{7,5} = 1$ $y_{7,6} = 3$ $y_{7,10} = 6$

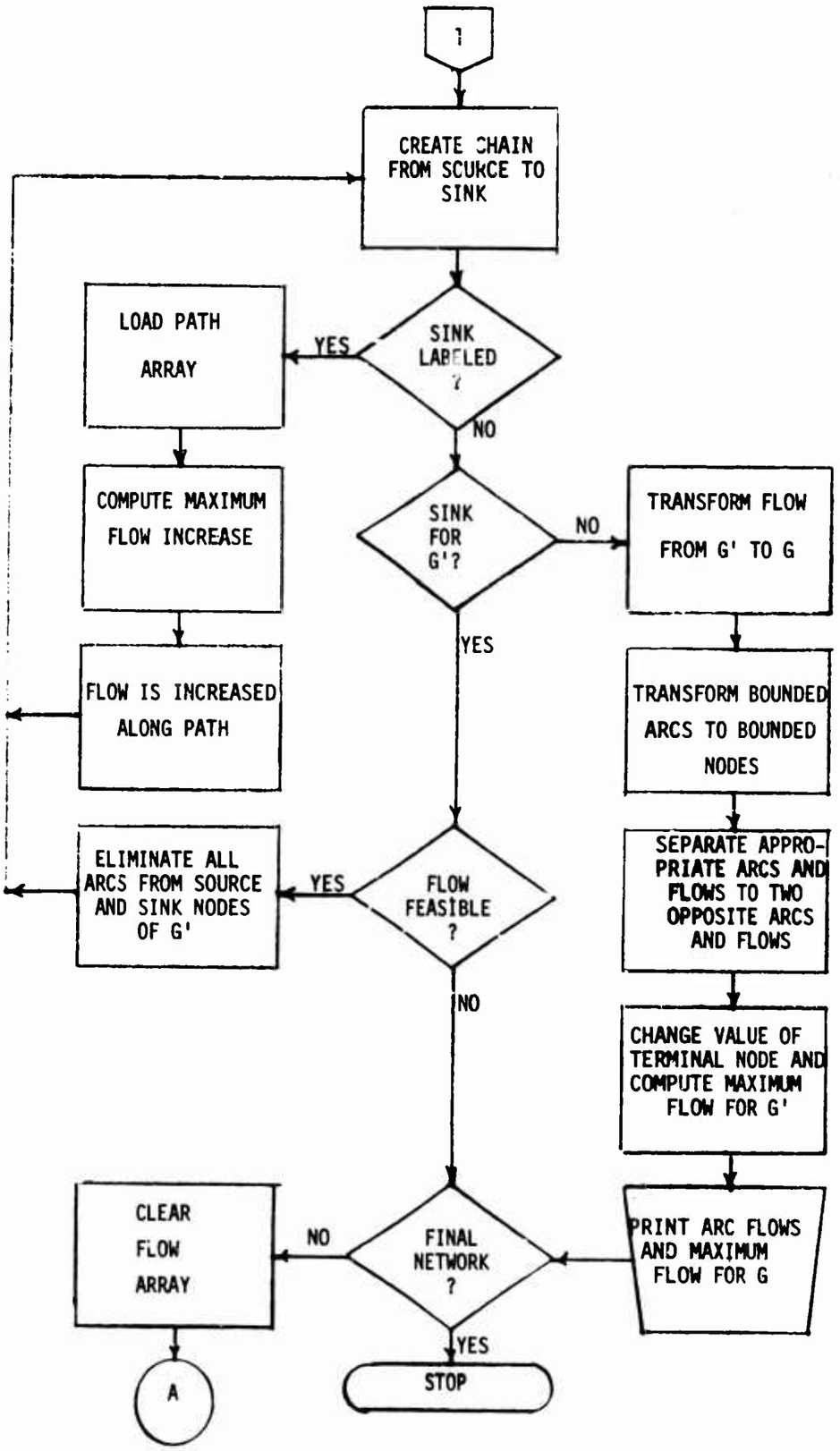
   The flow in node 5 is 9 and in node 6 is 9.

Computer Program Description

Since it is often the case that it is necessary to solve the maximal flow in a much larger and more complicated network, a computer program for the solution of the generalized maximal flow problem as presented in this thesis has been written. The program is written in FORTRAN IV for use on an IBM 360 computer.



GENERAL FLOWCHART



GENERAL FLOWCHART CONTINUED

The maximal flow in generalized networks is computed using the methodology outlined in this chapter. In addition, a set of routines is added to the program which will permit the user to transfer several networks to input data directly without first reducing the networks to basic networks by hand or auxiliary methods. However, as the program is written each network must have a single source and sink and all undirected arcs are to be represented by two oppositely directed arcs. The program has been checked out and functions properly.

The present program is written for a 50 node network. In order to solve larger networks the size of the arrays in the dimension statement must be increased. The general flow chart of the entire program is given on pages 30 and 31. For the interested reader a detailed flowchart is presented in Appendix A. A Fortran listing of the program is contained in Appendix B along with the solution to the previous example.

Conclusions and Recommendations

The general maximal flow problem can be solved more efficiently by use of the method developed by H. Greenberg. The advantages of the technique are evident when solving small problems by hand. The techniques presented provide a systematic approach to obtaining a solution. The problem of computing the maximal flow in a mixed network in which the flow in the undirected arcs are bounded below by a positive number cannot be solved by this method. In fact, to the best of my knowledge this problem is unsolved in the general case except by complete enumeration techniques.

BIBLIOGRAPHY

1. Claude Berge, The Theory of Graphs and Its Applications, John Wiley and Sons, 1958.

2. C. Berge and A. Ghouils-Houri, Programming, Games and Transportation Networks, John Wiley and Sons, 1965.

3. R. G. Busaker, T. H. Saaty, Finite Graphs and Networks, McGraw-Hill, 1965.

4. G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, 1963.

5. G. B. Dantzig, D. R. Fulkerson, On Max Flow Min Cut Theorem of Networks, Rand Research Memorandum, R M - 1418 - 1, 1955.

6. G. B. Dantzig, D. R. Fulkerson, Computation of Maximal Flow in Networks, Rand Research Memorandum, R M - 1489, 1955.

7. L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton University Press, 1962.

8. D. R. Fulkerson, A Network - Flow Feasibility Theorem and Combinatorial Applications, Canadian Journal of Mathematics, Vol. II (1959), pp. 440-451.

9. Gale, David, A Theorem on Flows in Networks, Pacific Journal of Mathematics, Vol. 7 (1957), pp. 1073-1082.

10. Greenberg, H., On Generalized Maximal Flow in Networks, unpublished.

11. Ford, L. R. and Fulkerson, D. R., Network Flow and Systems of Representatives, Canadian Journal of Mathematics, Vol. 10 (1958), pp. 78-85.

12. _____, A Simple Algorithm for Finding Maximal Network Flows and Applications to the Hitchcock Problem, Canadian Journal of Mathematics, Vol. 9 (1957), pp. 210-218.

APPENDIX A

DETAILED FLOW CHARTS

This appendix contains the detailed flow chart of the computer program in addition to a description of variables used. The statements are written in FORTRAN IV with arrows indicating the logical flow through the system. The input and output statements are not enclosed.

Description of Variables

Variable Name	Description
C(I,J)	Capacity of the directed arc from node I to node J.
B(I,J)	Lower bound of the directed arc from I to node J.
Y(I,J)	Initially used to read in data.
Y(I,1)	The number of the node the arc $a_{i,j}$ is incident from.
Y(I,2)	The number of the node the arc $a_{i,j}$ is incident to.
Y(I,3)	Upper bound of arc $a_{i,j}$.
Y(I,4)	Lower bound of arc $a_{i,j}$.
Y(I,J)	The value of the flow in a directed arc from I to J after input data is transmitted to arrays C(I,J) and B(I,J).
XL(I)	Dummy array used to compute values of the capacities of the source and sink arcs for G' and then the label of the nodes in the label routine.
NL(I)	Dummy array used in the label routine as a check to determine if all nodes have been checked and then to store the path used to increase flow.
XNC(I,1)	Number of the node that is bounded.
XNC(I,2)	Upper bound on a flow through a node.
XNC(I,3)	Lower bound on flow through a node.
XNC(I,4)	Value of flow through a node.
NN	Number of networks.

N	Number of nodes in the original problem.
M	Value of the designated number of the sink node.
KX	Value of the designated number of the source node.
NA	Number of arcs in the original problem.
NC	Number of bounded nodes.

Input Card Format

All entries on the card format in this section are right, justified and are fixed point entries if the variable name begins with I, J, K, L, M, or N and are floating point entries for all other variable names.

Card #1

Column	Name
1 - 4	NN

Card #2

1 - 4	N
4 - 8	NA
8 - 12	NC

Card #3 through Card NA + 1

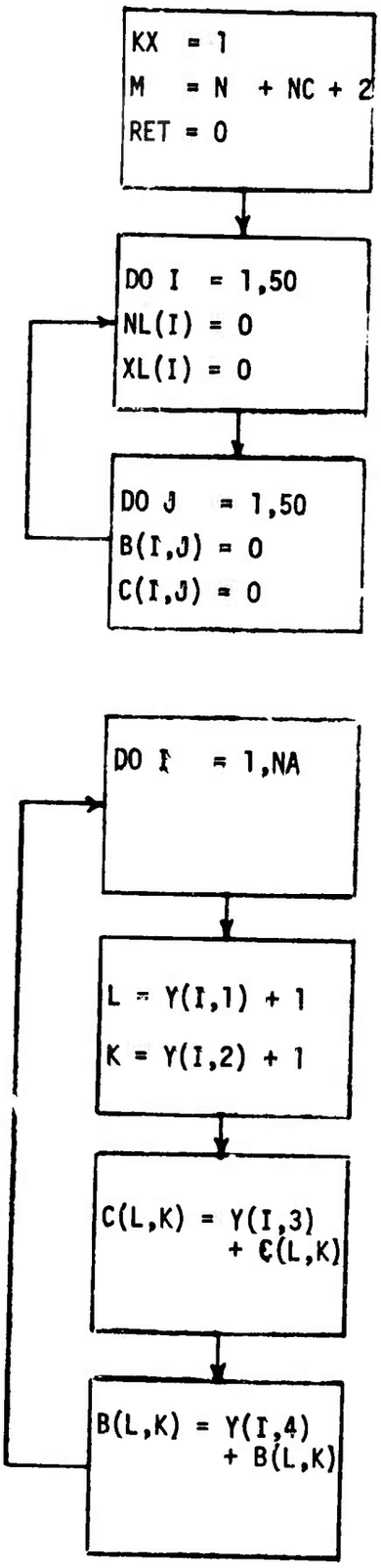
1 - 10	Y(I,1)
11 - 20	Y(I,2)
21 - 30	Y(I,3)
31 - 40	Y(I,4)

Card #NA + 3 through Card NC + 3

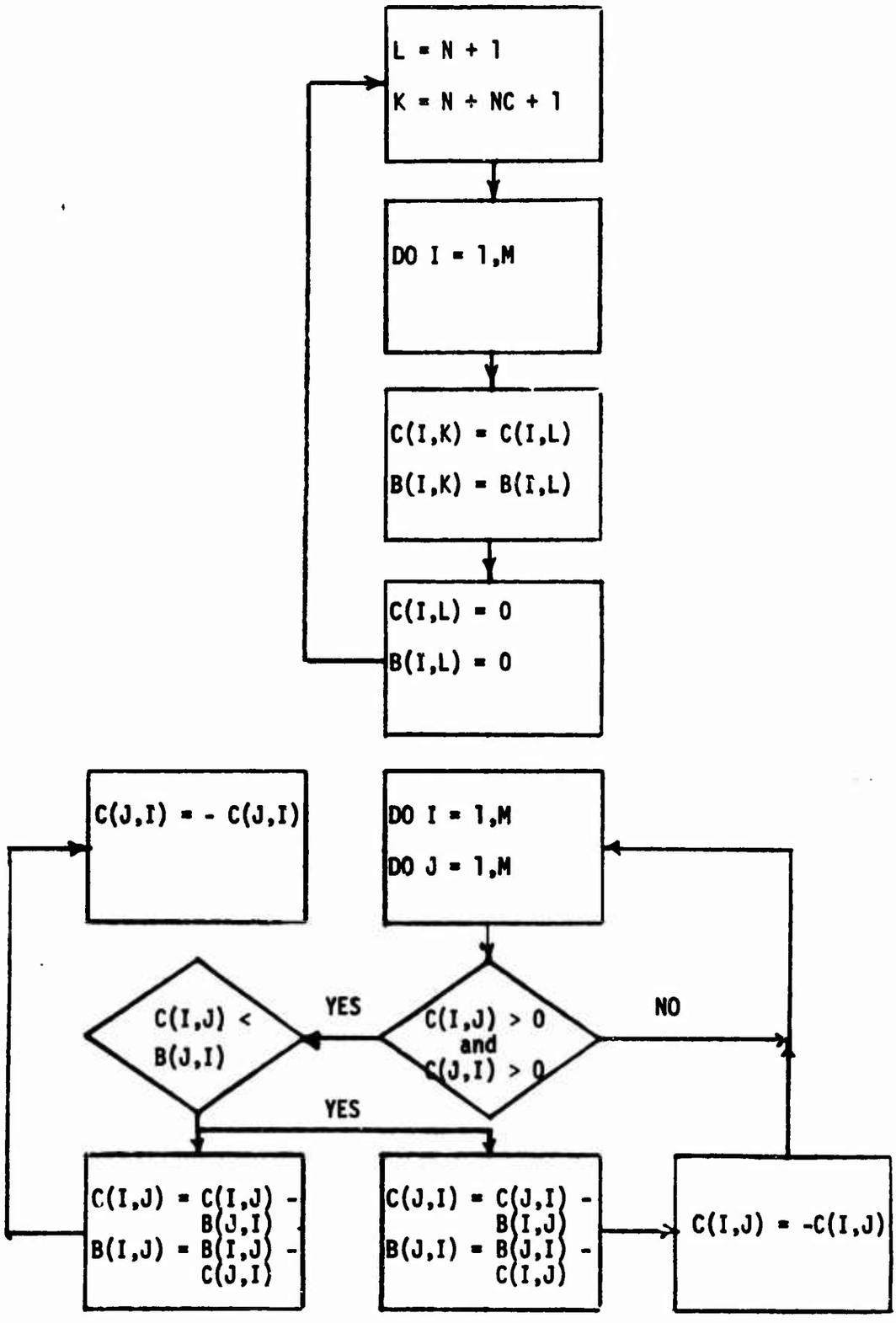
Column	Name
1 - 10	XMC(I,1)
11 - 20	XNC(I,2)
21 - 30	XNC(I,3)

Detailed Flowchart

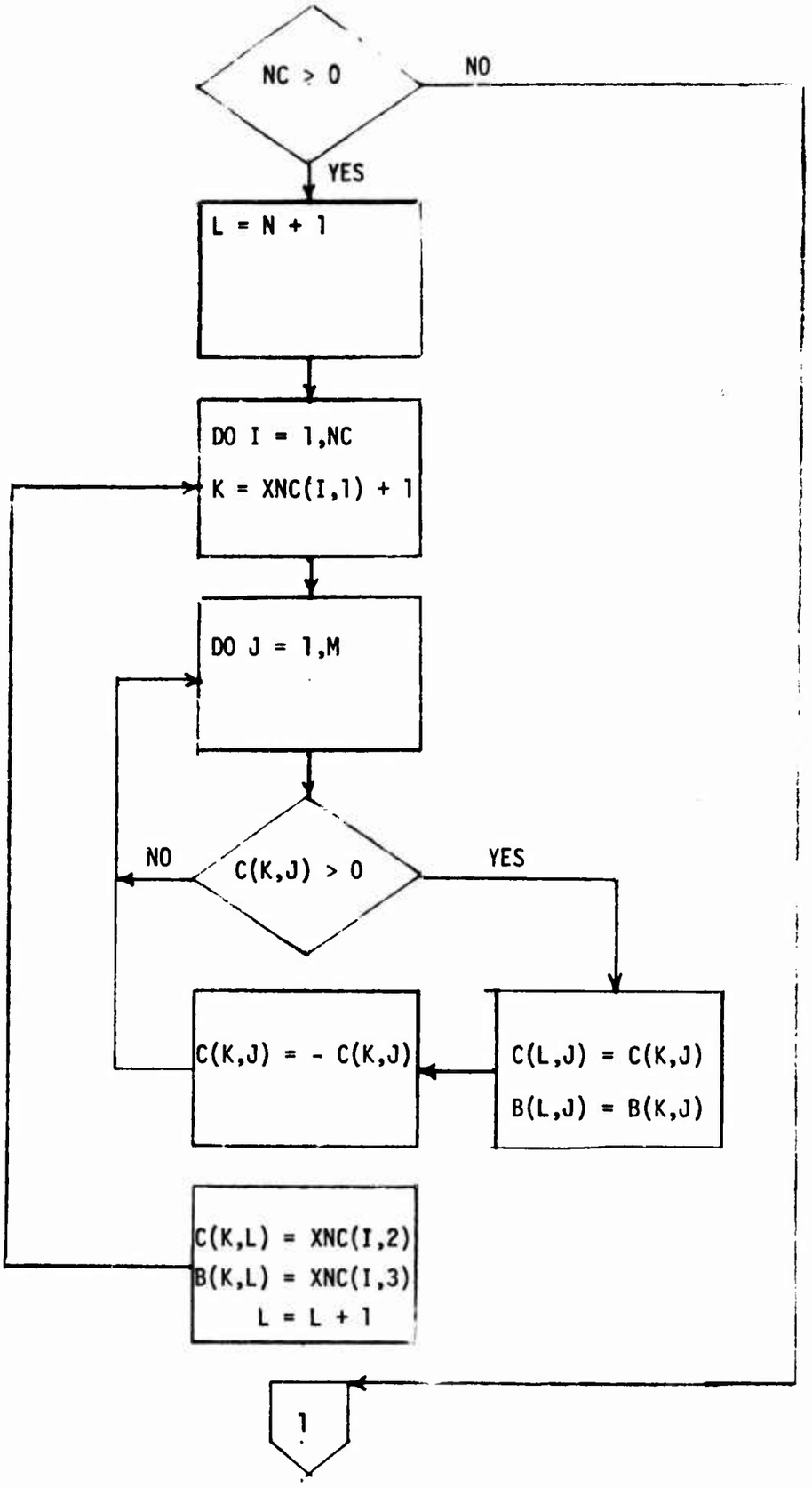
The detailed flowchart of the program is given in the next 11 pages.



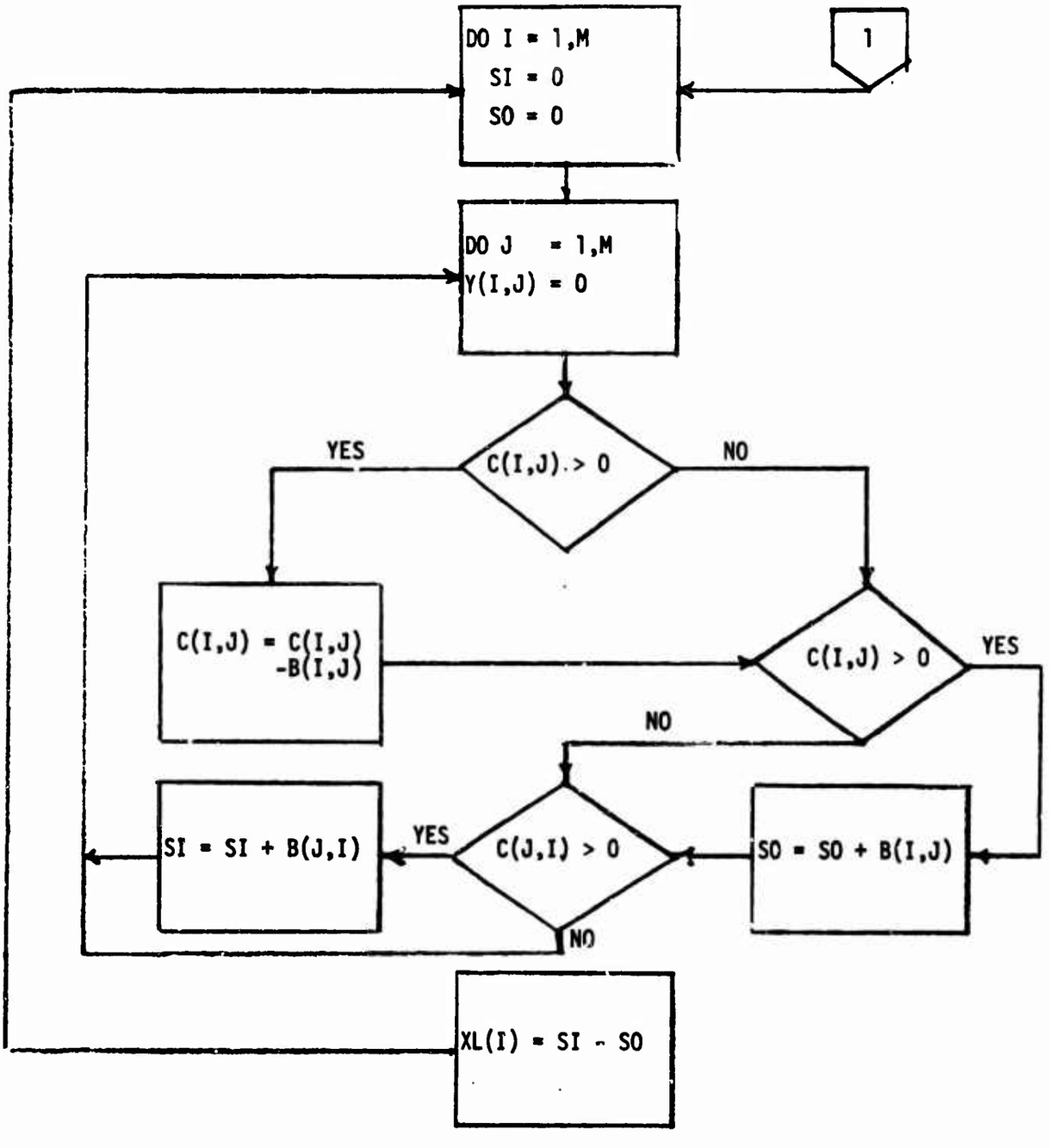
DETAILED FLOWCHART



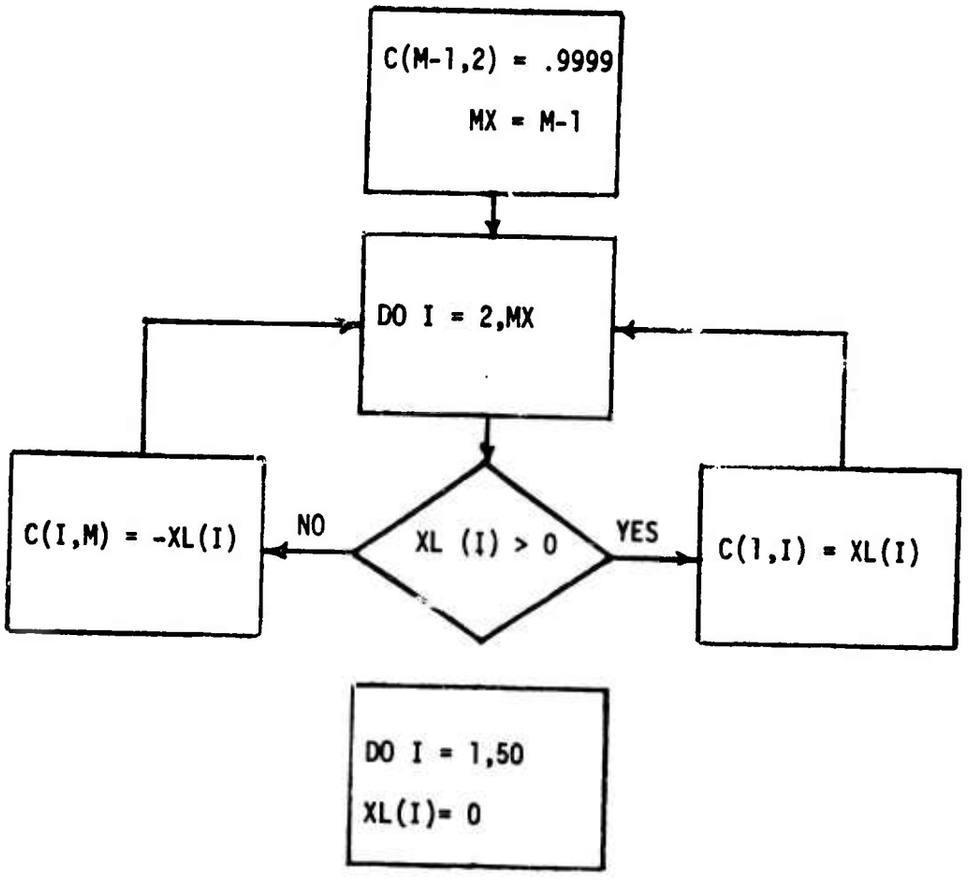
DETAILED FLOWCHART



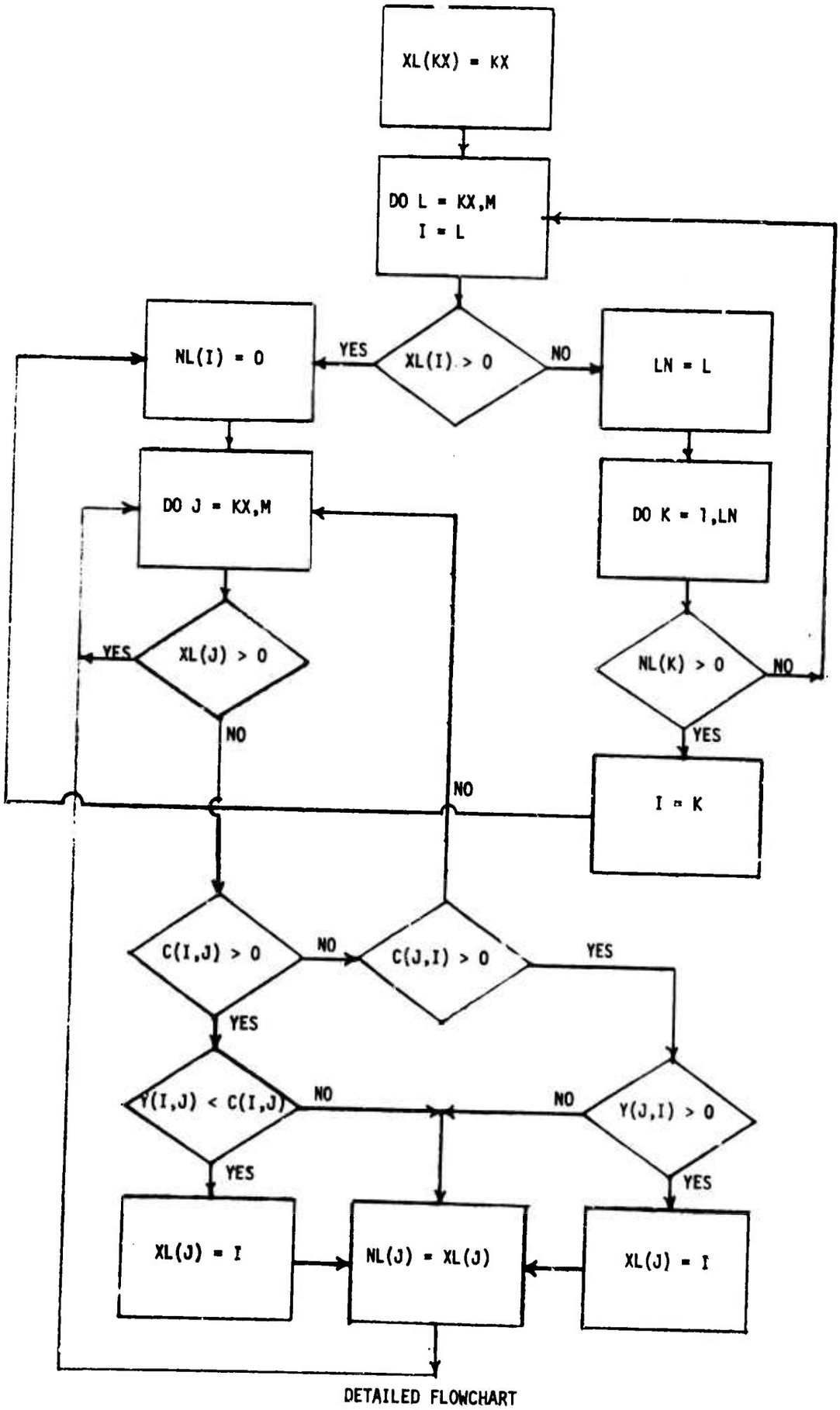
DETAILED FLOWCHART



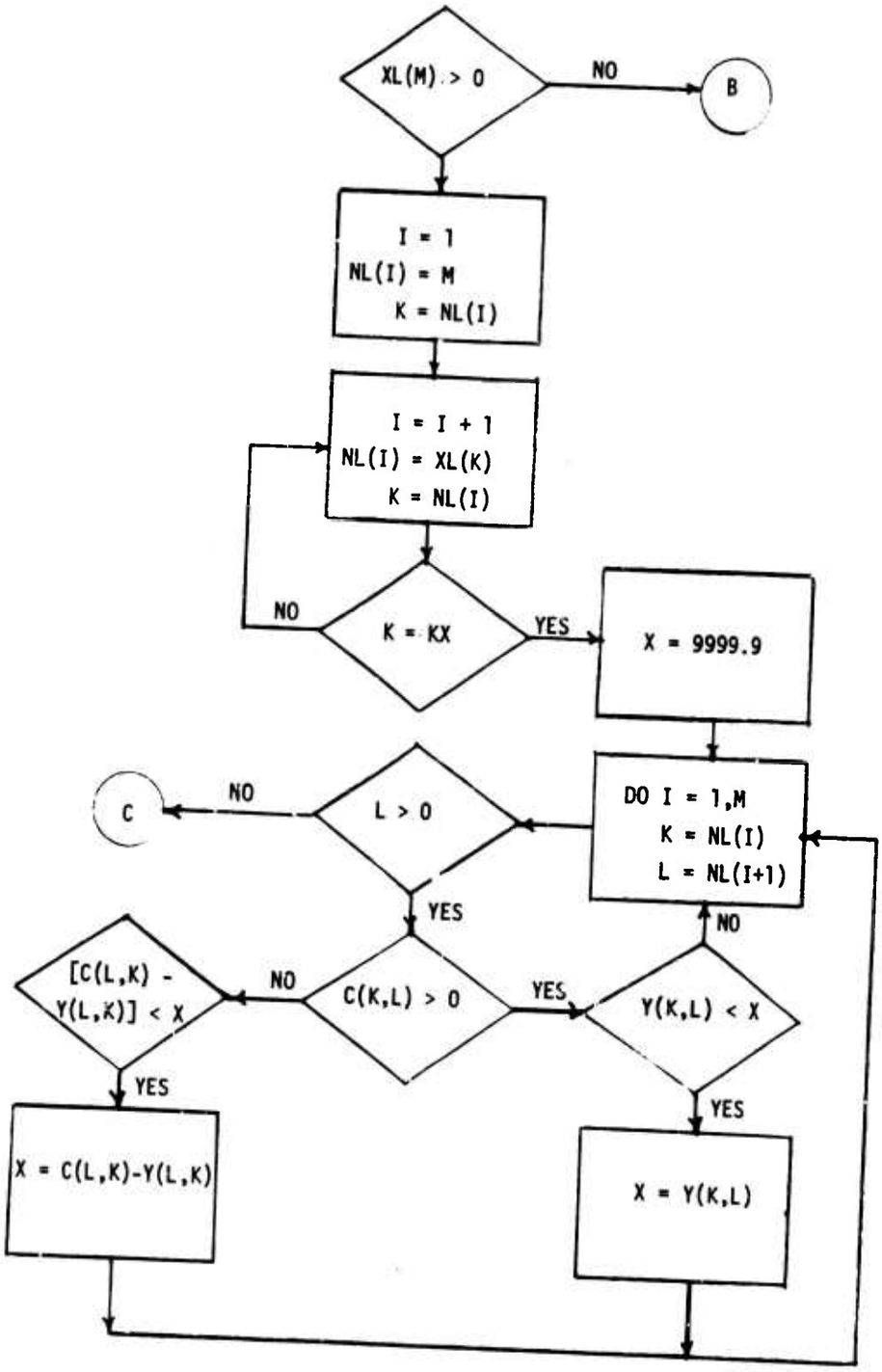
DETAILED FLOWCHART



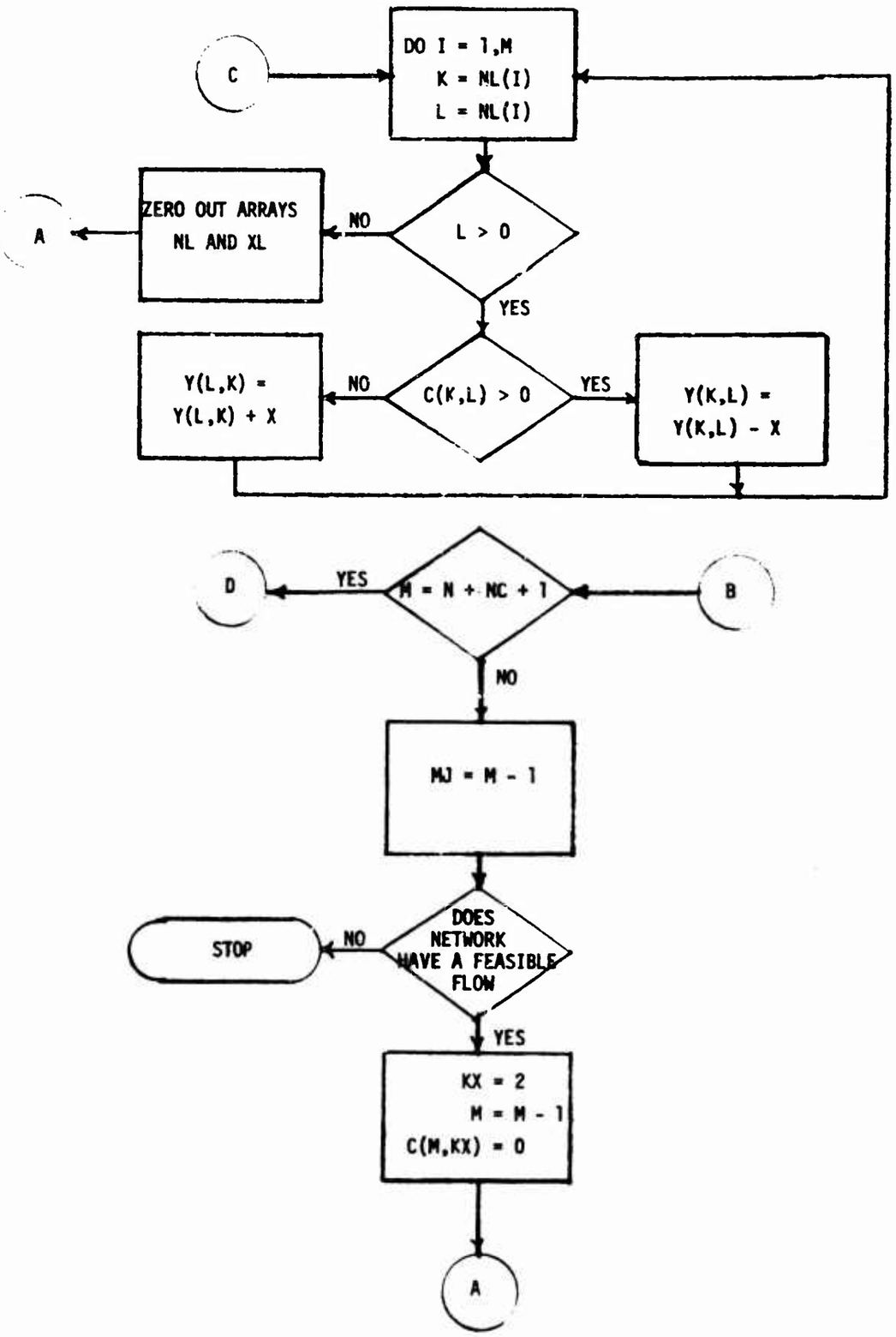
DETAILED FLOWCHART



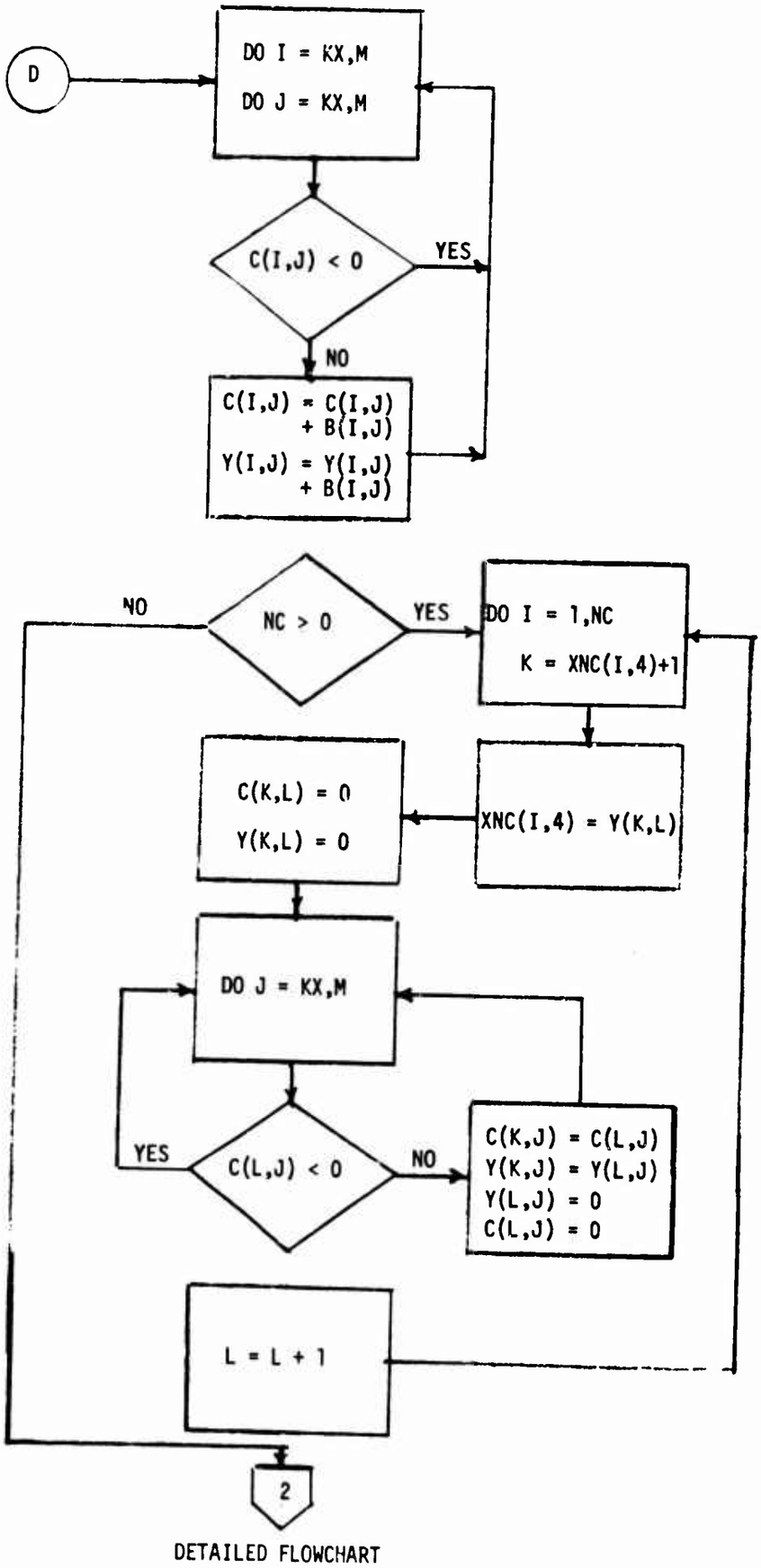
DETAILED FLOWCHART



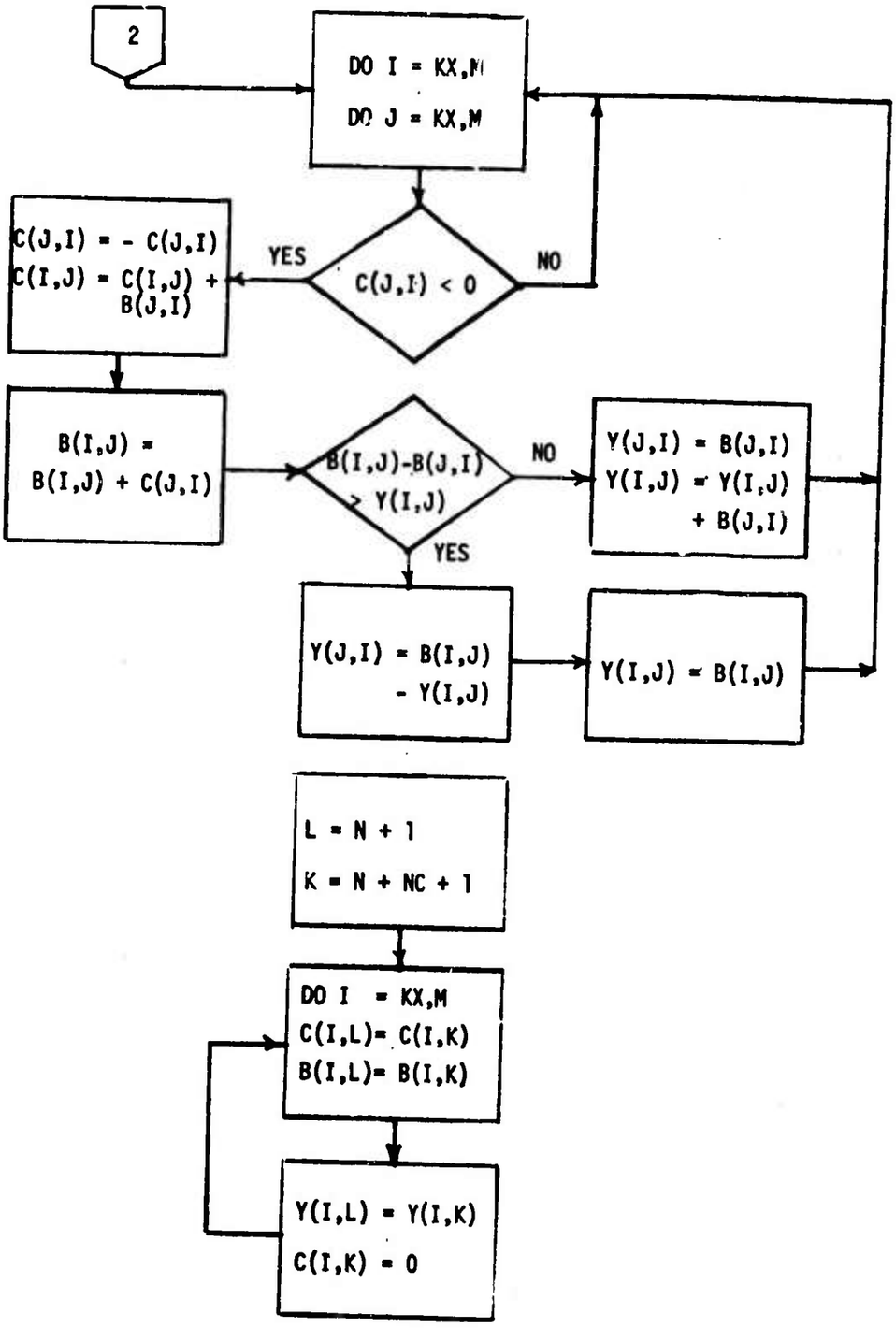
DETAILED FLOWCHART



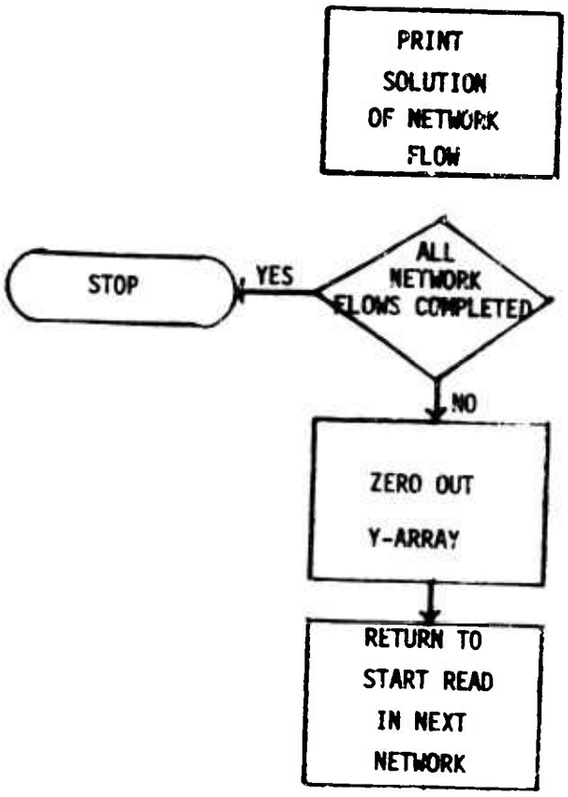
DETAILED FLOWCHART



DETAILED FLOWCHART



DETAILED FLOWCHART



DETAILED FLOWCHART

APPENDIX B

FORTRAN - IV LISTING

This appendix contains the complete listing of the computer program. The problem illustrated in Chapter IV was used to display the output format. Comment cards are included for clarification and identification of the main sections of the program.

```
      DIMENSION C(50,50),B(50,50),Y(50,50),XL(50),NL(50),
     1XNC(50,4)
      READ(5,801) NN
  801 FORMAT(I4)
      DO 800 IU=1,NN
      READ(5,1) N,NA ,NC
    1 FORMAT(3I4)
      DO 2 I=1,NA
      READ(5,3)( Y(I,J),J=1,4)
    3 FORMAT(4F10.4 )
    2 CONTINUE
      DO 8 I=1,NC
      READ(5,4)(XNC(I,J),J=1,3)
    4 FORMAT(3F10.4)
    8 CONTINUE
      KX=1
      M= N+NC+ 2
      RET=0
      DO 5 I=1,50
      NL(I)=0
      XL(I) =0
      DO 5 J=1,50
      B(I,J)=0
      C(I,J)=0
    5 CONTINUE
C THE CAPICITY MATRIX IS LOADED AND ALL ARCS INCIDENT TO
C AND INCIDENT FROM THE SAME NODES ARE COMBINED
      DO 6 I=1,NA
      L= Y(I,1) + 1.
      K= Y(I,2) + 1.
      C(L,K)= Y(I,3)+ C(L,K)
      B(L,K)= Y(I,4)+ B(L,K)
    6 CONTINUE
C  NUMBERICAL VALUE OF THE TERMINAL NODE IS CHANGED TO
C ALLOW FOR EXPANSION OF THE NETWORK
      L=N+1
      K=N+NC+1
      DO 11 I=1,M
      C(I,K)=C(I,L)
      B(I,K)=B(I,L)
      C(I,L)=0.
   11 B(I,L)=0.
C COMBINE MULTIPLE ARCS JOINING  TWO  NODES IN OPPOSITE
C DIRECTIONS.
      DO 15 I=1,M
      DO 15 J=1,M
      IF(C(I,J).GT. 0..AND. C(J,I).GT. 0.) GO TO 16
      GO TO 15
   16 IF(C(I,J).LE.B(J,I)) GO TO 17
      C(I,J)= C(I,J)-B(J,I)
      B(I,J)= B(I,J)-C(J,I)
      C(J,I)=-C(J,I)
      GO TO 15
   17 C(J,I)=C(J,I)- B(I,J)
      B(J,I)=B(J,I)- C(I,J)
```

```
      C(I,J)=-C(I,J)
   15 CONTINUE
C TRANSLATE  BOUNDED NODES INTO BOUNDED ARCS
      IF(NC)155,155,156
  156 L=N+1
      DO 34 I=1,NC
      K= XNC(I,1)+1.
      DO 32 J=1, M
      IF(C(K,J)) 32,32,33
   33 C(L,J)= C(K,J)
      B(L,J)= B(K,J)
      IF(C(K,J) .GT.0.) C(K,J)=-C(K,J)
   32 CONTINUE
      C(K,L)= XNC(I,2)
      B(K,L)=XNC(I,3)
      L=L+1
   34 CONTINUE
  155 CONTINUE
C THE FOLLOWING OPERATION COMPUTES THE CAPACITIES OF THE
C ARCS FROM AND TO THE SOURCE AND SINK NODES IN G'
      DO 9 I=1,M
      SI=0
      SO=0
      DO 10 J=1,M
      Y(I,J)=0
      IF(C(I,J).GT. 0.) C(I,J)= C(I,J)-B(I,J)
      IF(C(I,J).GT.0.) SO=SO+B(I,J)
      IF(C(J,I).GT.0) SI=SI+B(J,I)
   10 CONTINUE
      XL(I)=SI-SO
    9 CONTINUE
C G'IS COMPLETED BY ADDING NEW SINK AND SOURCE ARCS
      C(M-1,2)=9999
      MX= M-1
      DO 12 I=2, MX
      IF( XL(I)) 13,12,14
   13 C(I,M)= -XL(I)
      GO TO 12
   14 C(1,I)= XL(I)
   12 CONTINUE
      DO 31 I=1,50
   31 XL(I)=0
   20 XL(KX)=KX
C THE LABEL ROUTINUE AS  PRESENTED  BY FORD AND FULKERSON
C IS USED TO DETERMINE THE FLOW CAN BE INCREASED
      DO 210 L=KX,M
      I=L
      IF(XL(I))93,93,96
   96 NL(I)=0
      GO TO 22
   93 LN=L
      DO 94 K=1,LN
      IF(NL(K))94,94,95
   95 I=K
      GO TO 96
```

```
   94 CONTINUE
      GO TO 210
   22 DO 21 J= KX, M
      IF(XL(J))23,23,21
   23 IF(C(I,J)) 24,24,25
   25 IF( Y(I,J).LT. C(I,J)) XL(J)=I
      NL(J)=XL(J)
      GO TO 21
   24 IF(C(J,I))21,21,26
   26 IF(Y(J,I).GT. 0.)XL(J)=I
      NL(J)=XL(J)
   21 CONTINUE
      GO TO 93
  210 CONTINUE
      IF(XL(M))27,27,28
C A PATH FROM   SOURCE TO  SINK  IS  DETERMINED DIRECTLY
C FROM THE LABELLED NODES
   28 I=1
      NL(I)=M
      K=NL(I)
   30 I=I+1
      NL(I)=XL(K)
      K=NL(I)
      IF( K .EQ. KX)GO TO 40
      GO TO 30
   40 X=9999.9
C THE MAXIMUM  AMOUNT THAT THE FLOW CAN BE INCREASED
C ALONG THE PATH IS COMPUTED
   45 DO 41 I=1,M
      K=NL(I)
      L=NL(I+1)
      IF(L)50,50, 42
   42 IF(C(K,L))43,43,44
   43 IF((C(L,K)- Y(L,K)).LT. X) X= C(L,K)-Y(L,K)
      GO TO 41
   44 IF( Y(K,L).LT. X) X= Y(K,L)
   41 CONTINUE
   50 CONTINUE
C THE FLOW IS INCREASED
      DO 51 I=1,M
      K=NL(I)
      L=NL(I+1)
      IF(L)60,60, 52
   52 IF(C(K,L)) 53,53,54
   53 Y(L,K)= Y(L,K)+ X
      GO TO 51
   54 Y(K,L)= Y(K,L) -X
   51 CONTINUE
   60 CONTINUE
      DO 61 I= KX,M
      NL(I)=0
   61 XL(I)=0
      GO TO 20
C THE SINK CANNOT BE LABELLED AND A CHECK IS MADE
C TO DERMINE IF THE FLOW IS FEASIBLE
```

```
   27 IF(M.EQ.(N+NC+1)) GO TO 80
      MJ=M-1
      DO 70 I= 2, MJ
      IF((C(1,I)- Y(1,I)) .GT. .0001 .OR. (C(I,M)-
     1Y(I,M)).GT. .0001) GO TO 90
   70 CONTINUE
      KX=2
      M=M-1
      C(M,KX)=0
      GO TO 20
C THE ORIGINAL NETWORK IS RECONSTRUCTED  AND FLOW IS
C TRANSFERED FROM G' TO G
   80 CONTINUE
      DO 81 I=KX,M
      DO 81 J=KX,M
      IF(C(I,J).LT.0.) GO TO 81
      C(I,J)= C(I,J) +B(I,J)
      Y(I,J)=Y(I,J)+B(I,J)
   81 CONTINUE
      IF(NC)157,157,158
  158 L=N+1
      DO 100 I=1,NC
      K= XNC(I,1)+1.
      XNC(I,4) = Y(K,L)
      C(K,L)=0.
      Y(K,L)=0
      DO 101 J=KX,M
      IF(C(L,J).LE.0) GO TO 101
      C(K,J)=C(L,J)
      Y(K,J)=Y(L,J)
      Y(L,J)=0
  101 C(L,J)=0
      L=L+1
  100 CONTINUE
  157 CONTINUE
      DO 102 I= KX,M
      DO 102 J= KX,M
      IF(C(J,I))103,102,102
  103 C(J,I)=-C(J,I)
      C(I,J)= C(I,J)+B(J,I)
      B(I,J)= B(I,J)+C(J,I)
      IF((B(I,J)-B(J,I)).GT. Y(I,J)) GO TO 104
      Y(J,I)=B(J,I)
      Y(I,J)=Y(I,J)+B(J,I)
      GO TO  102
  104 Y(J,I)=B(I,J)-Y(I,J)
      Y(I,J)=B(I,J)
  102 CONTINUE
      L=N+1
      K=N+NC+1
      DO 105 I=KX,M
      C(I,L)=C(I,K)
      B(I,L)=B(I,K)
      Y(I,L)=Y(I,K)
  105 C(I,K)=0.
```

```
      WRITE(6,86)
   86 FORMAT(1H1, 25X, 'ARC',9X,'LOWER BOUND',12X,'FLOW',
     114X,'UPPER BOUND',//)
      DO 82 I=2,M
      RET=RET+Y(I,M)
      DO 82 J=2,M
      IF(C(I,J))82,82,85
   85 L=I-1
      K=J-1
      WRITE(6,83) L,K,B(I,J),Y(I,J),C(I,J)
   83 FORMAT(24X,I2,1X,I2, 5X,F10.3,11X,F10.3,10X,F10.3,/)
   82 CONTINUE
      WRITE(6,87)
   87 FORMAT(1HO,24X,'NODE',9X,'LOWER BOUND',12X,'FLOW',
     114X,'UPPER BOUND',//)
      IF(NC)161,161,160
  160 DO 89 I=1,NC
      K=XNC(I,1)
      WRITE(6,88)     K     ,XNC(I,3),XNC(I,4),XNC(I,2)
   88 FORMAT(25X,I2,6X,F10.3,11X,F10.3,10X,F10.3,//)
   89 CONTINUE
  161 CONTINUE
      WRITE(6,84)  RET
   84 FORMAT( 25X,'MAX FLOW =',F10.4)
      GO TO 91
   90 WRITE (6,92)
   92 FORMAT(25X,'NETWORK HAS NO FEASIBLE FLOW')
   91 CONTINUE
      DO 117 I=1,50
      DO 117 J=1,50
      Y(I,J)=0
  117 CONTINUE
  800 CONTINUE
      END
```

ARC	LOWER BOUND	FLOW	UPPER BOUND
1 2	0.0	12.000	999.000
1 3	0.0	8.000	999.000
2 4	8.000	8.000	13.000
2 5	0.0	4.000	4.000
3 4	3.000	3.000	7.000
3 5	5.000	5.000	10.000
4 5	3.000	7.000	13.000
4 6	1.000	7.000	7.000
5 4	3.000	3.000	13.000
5 6	0.0	0.0	8.000
5 7	3.000	7.000	7.000
5 8	2.000	5.000	5.000
6 4	0.0	0.0	2.000
6 5	0.0	2.000	8.000
6 7	1.000	3.000	3.000
6 8	3.000	9.000	9.000
7 5	1.000	1.000	4.000
7 6	3.000	3.000	8.000
7 8	4.000	6.000	10.000

NODE	LOWER BOUND	FLOW	UPPER BOUND
5	4.000	9.000	15.000
6	4.000	9.000	10.000

MAX FLOW = 20.0000

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
Computation of Maximum Flows in Networks

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Master's Thesis

5. AUTHOR(S) (Last name, first name, initial)
Burns, William C., Captain, USA

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1968	55	12

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES
This document is subject to special export controls and each transmittal to foreign nationals may be made only with prior approval of the Naval Postgraduate School.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A review of the current theory and methods for the computation of maximum flow in networks is presented along with a simplified method for determination of a feasible flow in networks with upper and lower bounded arcs. A computational procedure is presented which is used to calculate the maximum flow for a general network. The network is reduced to an equivalent basic network. An associated network is used to compute a feasible, then the maximum flow for the basic network. A computer program is included for use in computation of maximal flows in large networks.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Transportation Networks						
Network Flows						
Maximal Network Flow						